# ET-AVR JTAG (RS232) V1.0

**ET-AVR JTAG (RS232) V1.0** which is an ETT Board is designed to download HEX File and debug program into MCU AVR family of Atmel through JTAG Interface. It can use with JTAG Interface MCU only and must use with Program AVR Studio 4.XX.

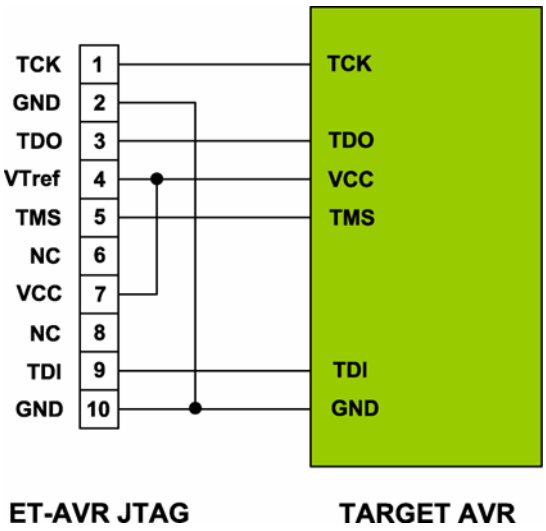
## Specifications of Board ET-AVR JTAG (RS232) V1.0

1. Its specifications are the same as AVR JTAG ICE from ATMEL.
2. Support Real Time debugging.
3. There's JTAG Interface Module for programming and debugging.
4. Be able to upgrade Firmware through Program AVR Studio 4 for using with new MCU and Firmware is attached with Program AVR Studio 4.
5. Be able to use with Power Supply between 2.7V-5.5V.
6. Use pressure from Target Board.
7. Communication through RS232 Serial Port.
8. There's LED Dislay operation status of Power, Active.

## AVR Microcontroller No. that can use with ET-AVR JTAG (AVR Studio 4.12)

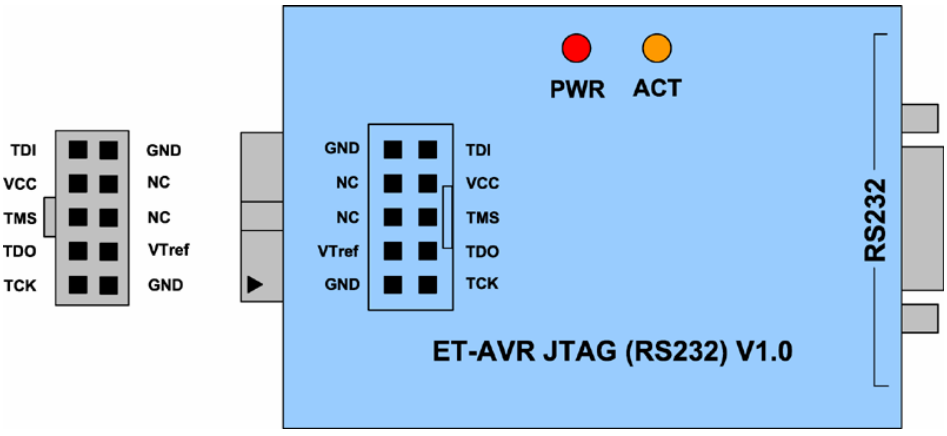
- ATmega16 , ATmega16L
- ATmega162 , ATmega162L
- ATmega169 , ATmega169L , ATmega169V
- ATmega32 , ATmega32L
- ATmega323 , ATmega323L
- ATmega64 , ATmega64L
- ATmega128 , ATmega128L
- AT90CAN128

## Connection between ET-JTAG AVR and AVR Microcontroller

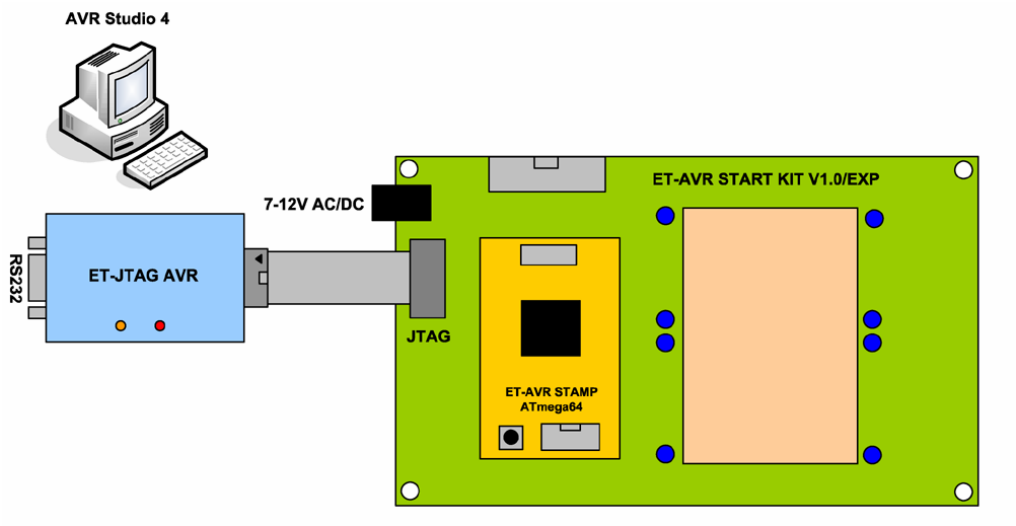



Picture displays the connection between ET-AVR JTAG with AVR Microcontroller.

It uses only cable of **TCK, TDO, TMS, TDI, VCC, GND** for connection. It is not necessary to interface in the part of VTref Pin because circuit of this ET-JTAG AVR pin is connected with VCC.




Picture displays Pin position of ET-AVR JTAG.

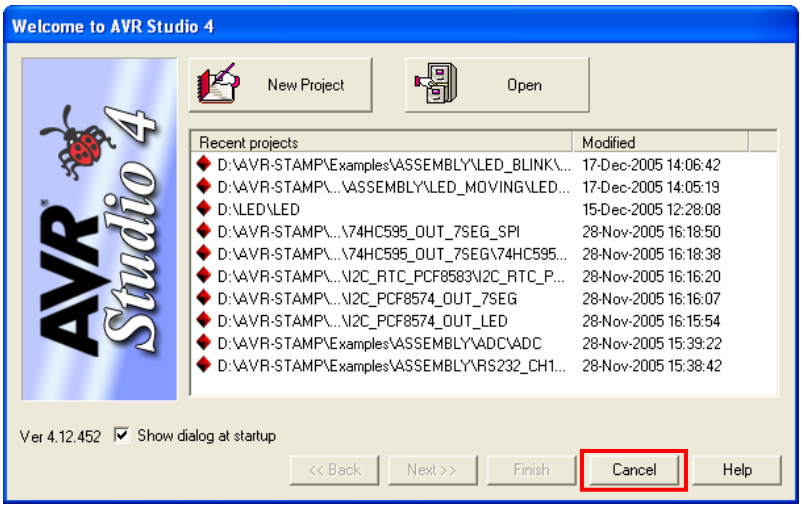



Picture displays the connection between ET-AVR JTAG and ET-AVR START KIT V1.0/EXP.

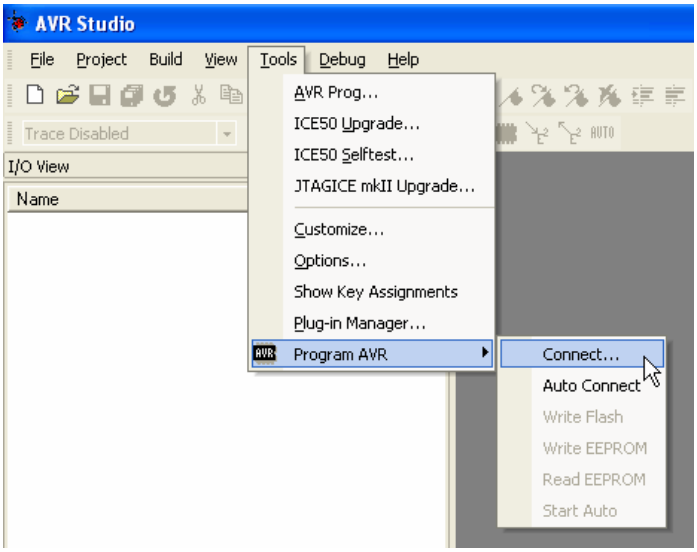
## Using ET-AVR JTAG in Mode Programming

User can assign **Security Bits** and **Configuration Bits** in this Mode as same as Program PonyProg2000 and its method to use this program is;

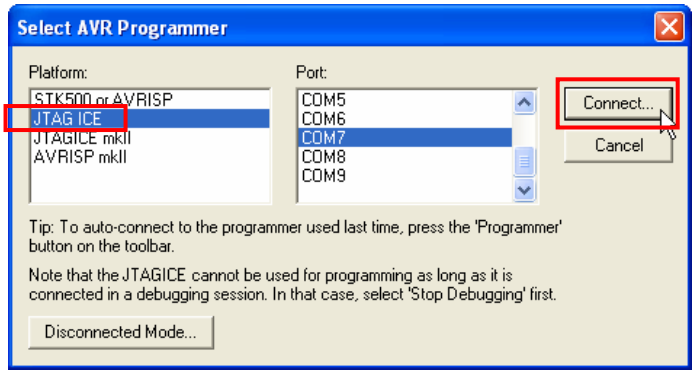
1. Open **program AVR Studio** and it will display Window of **Welcome to AVR Studio**, click **Cancel** to close this window.



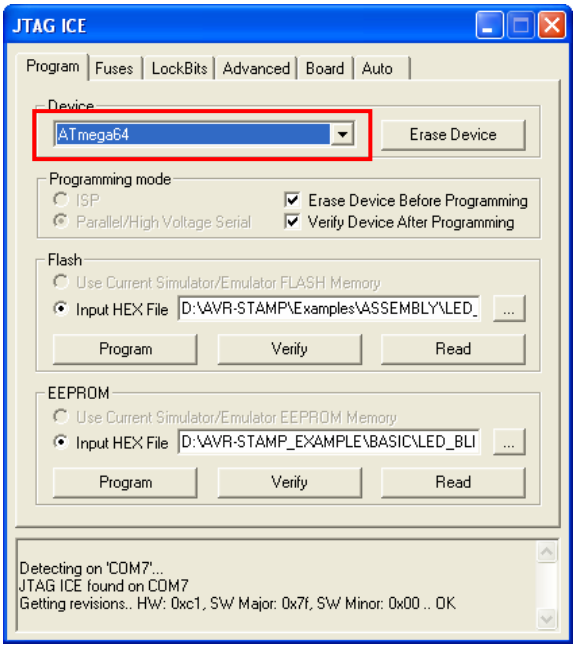
2. Select Menu **Tools → Program → AVR Connect…** as in the picture.



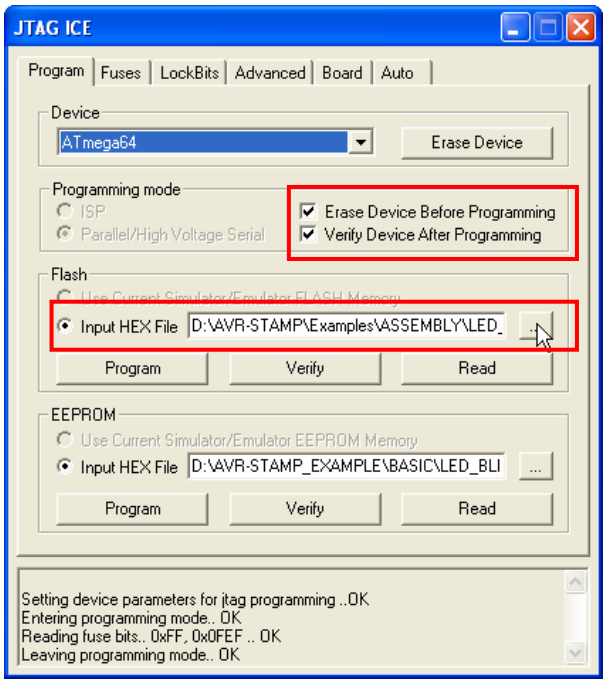

3. After that it will display window of **Select AVR Programmer** as in the picture, select **Platform** as **JTAG ICE** and **Port** that is connected with ET-AVR JTAG and then click **Connect**…

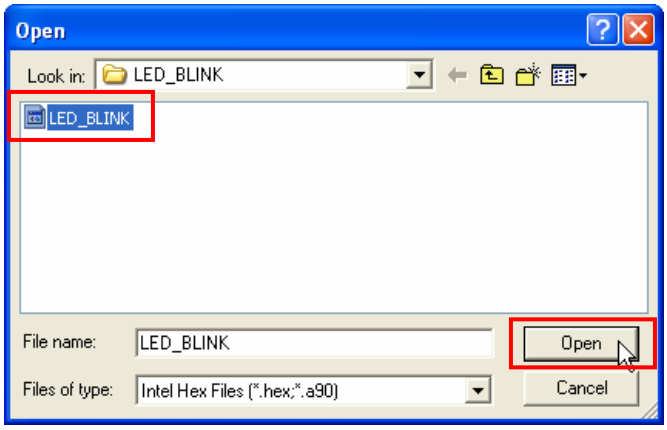



4. If program AVR Studio can connect with ET-JTAG AVR, it will display window of JTAG ICE as in the picture. Select MCU No. from blank of **Device**, in this case, we select as **ATmega64**.

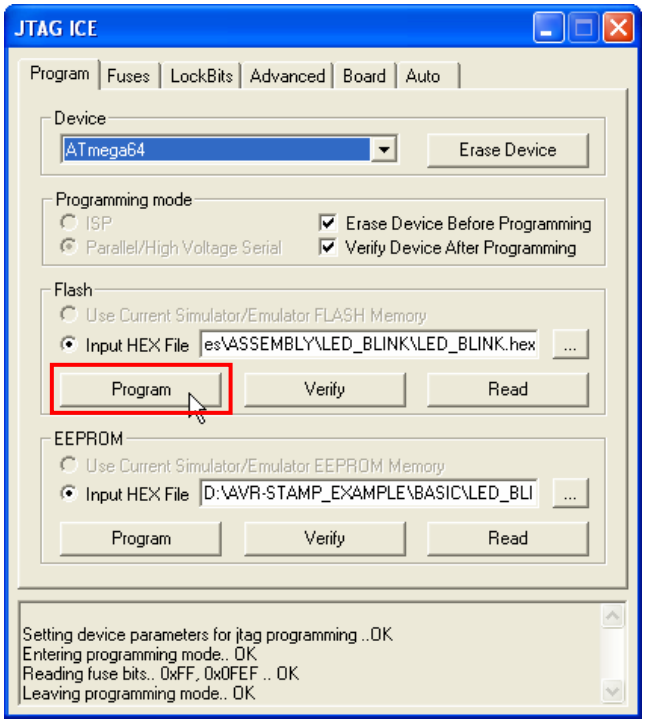



5. Open File for Programming into CPU, select them from blank of **Input HEX File** and specify name and address of HEX File completely. In blank of **Programming Mode**, select as **Erase Device Before Programming**, it uses to erase data before programming and select as **Verify Device After Programming**, it uses to verify data correctly or not after programming completely.

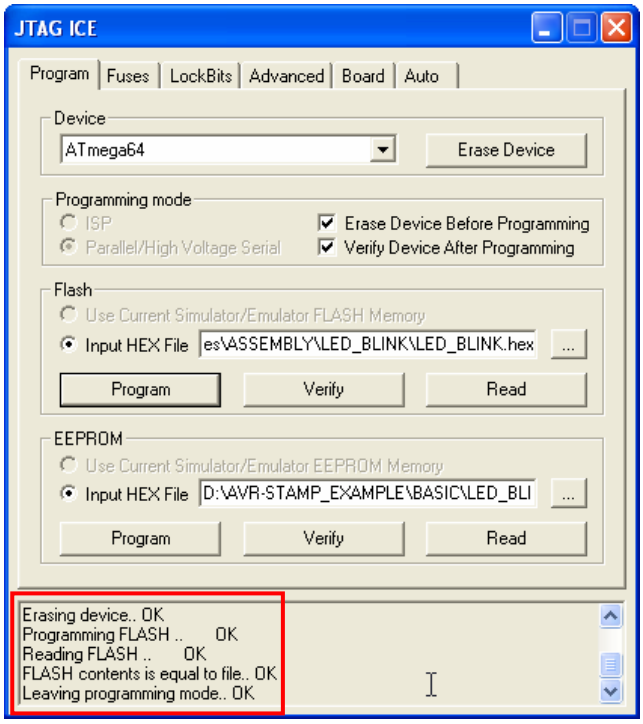




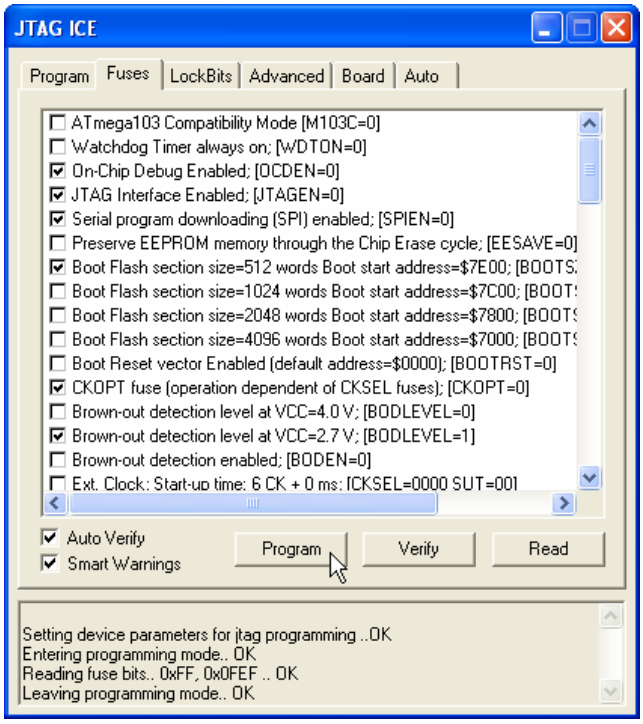

6. In case of not to adjust any default values of Fuses and LockBits, can press **Program** button to program Hex File into MCU instantly. Because default values of Fuses and LockBits is programmed before, so it are not erased with program. If user want to adjust default values of Fuses and LockBits, can adjust them after without any problem with program.

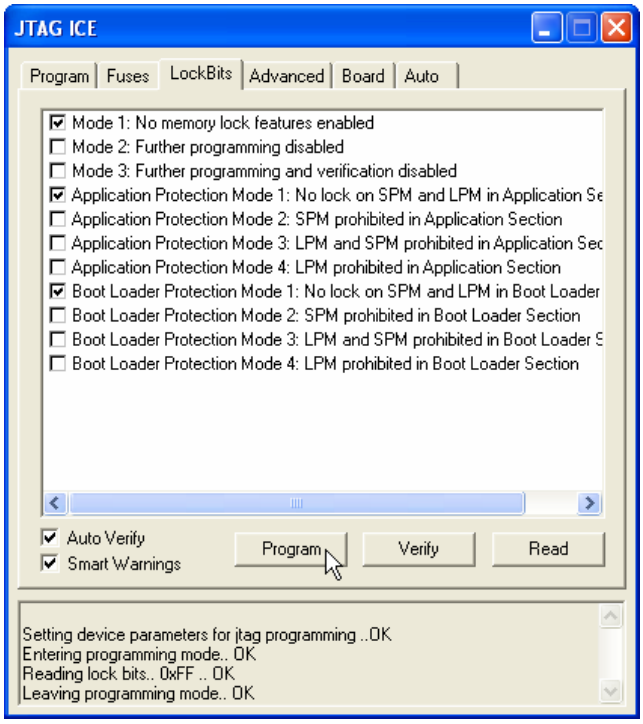


7. If there's no any misstake of programming Hex File into MCU, it will display the message as in the picture.




8. If user want to adjust default value of Fuses and LockBits, can select Fuses and LockBits to setup and program after as in the picture.






***NOTICE***

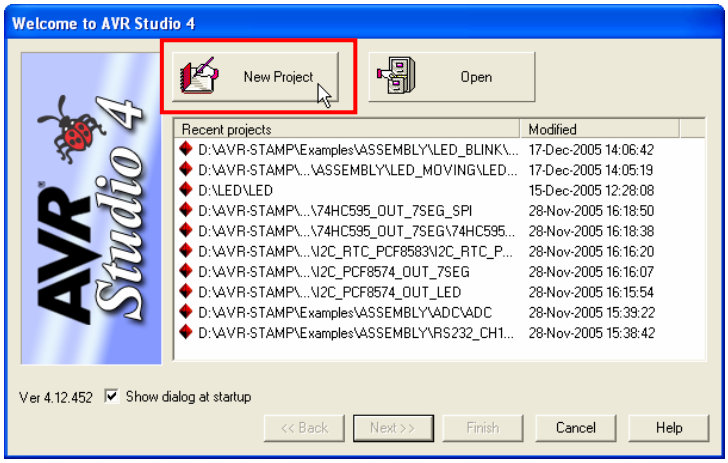
In case of using Board to experiment, default value of LockBits are not necessary to program because it uses to protect MCU from reading and rewriting again.

## Using ET-AVR JTAG in Mode Debugging

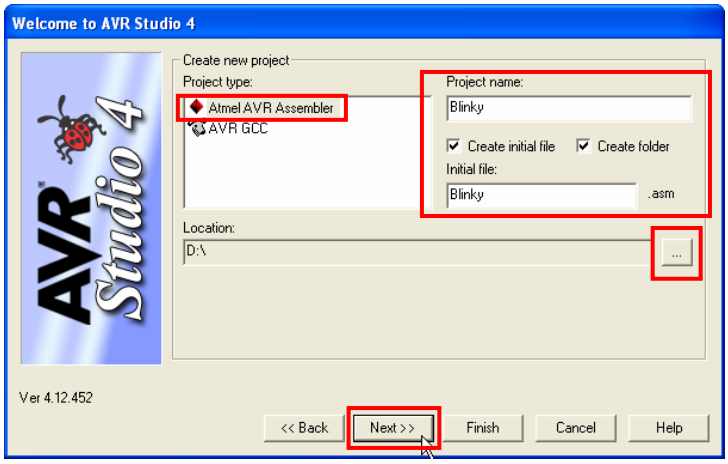
This mode uses to debug operation status of MCU. User can select as **Step by Step Debugging** or **Auto Debugging**. While debugging, default values of MCU will change as programming, so user can see its MCU default value instantly. It can debug with both Assembly Language and C Language. For example, if writing program is blinker, you can see blinker as status of debugging. Proceeding to use Mode debugging as following and in this case, we use one blinker to see easily.

# Exmaple Of Debugging with Assembly Language

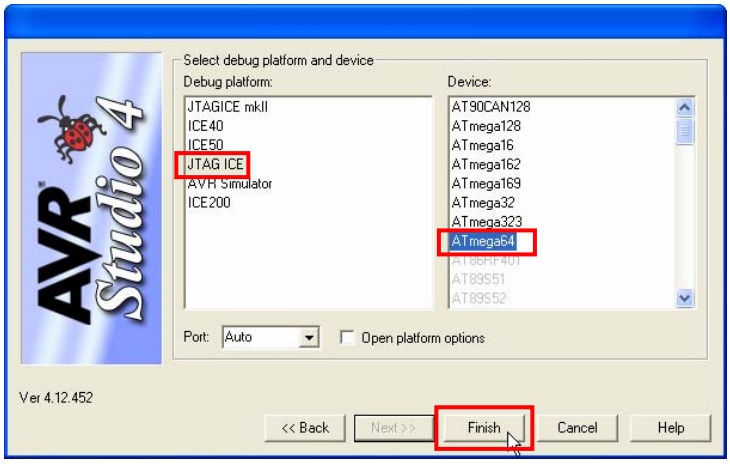
1. Open Program AVR Studio, it will display window of **Welcome to AVR Studio** and click **New Project** to create new project as in the picture.



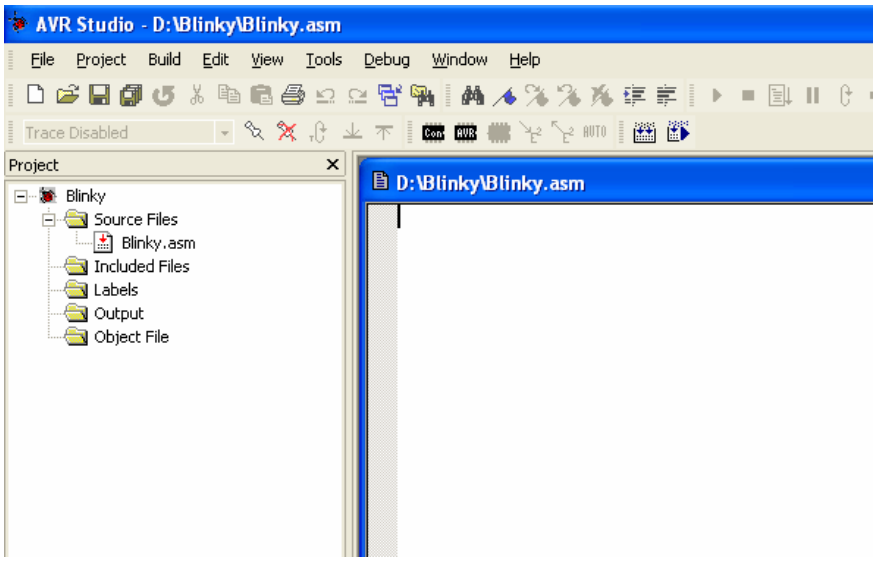

2. Select **Project Type** as **Atmel AVR Assembler** for writing program with Assembly Language. Assign Project name in blank of **Project name**, select blank of **Create initial file** to create File Assembly with create File project, and then select **Create Folder** to create new folder for saving File project. After that select directory to save file project and then click **Next** as in the picture.



3. Select **Debug Platform** as **JTAG ICE** and **Device** as ATmega64 and then click **Finish** as in the picture.



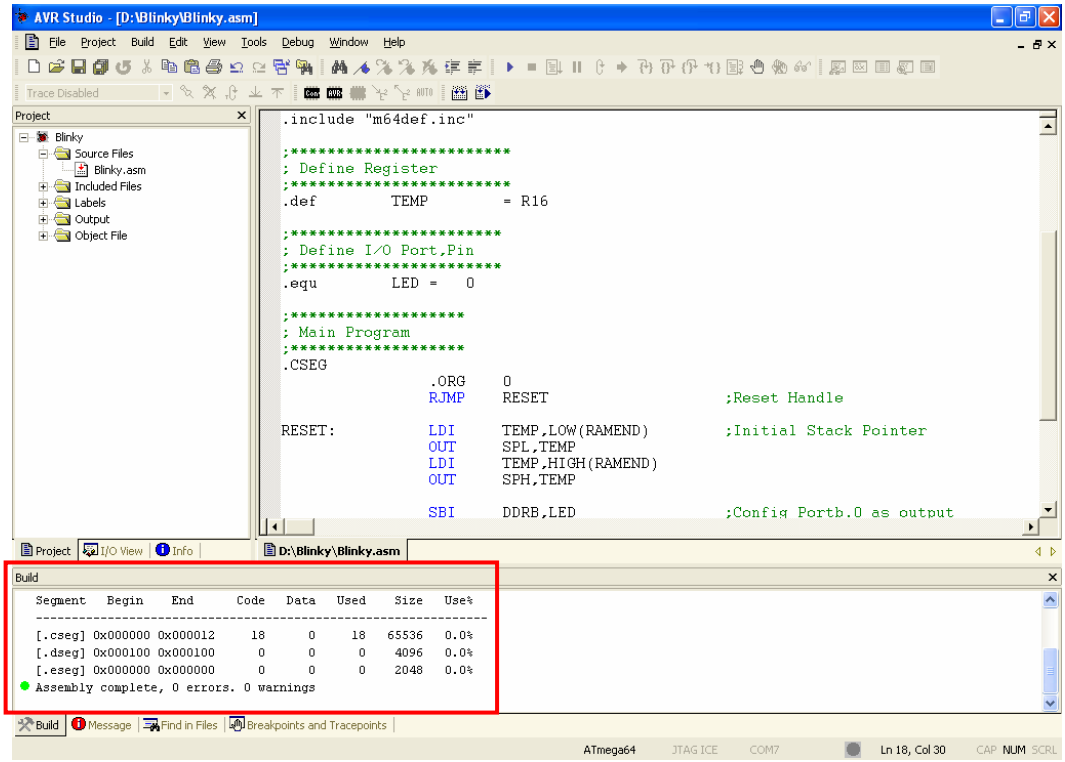

4. It will display window of **Text Editor** for writing program as in the picture.



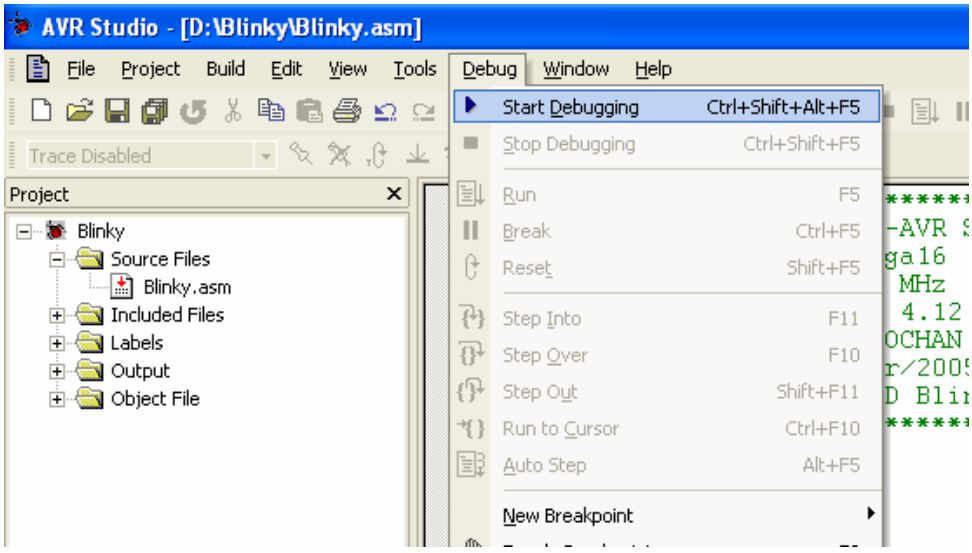

5. Type an example Assembly program as in the sample. In this case, we do not use program of time delay because we can see debugging instantly without time delay.

```
;*******************************************************
;* Examples Program For "ET-AVR STAMP ATmega64" Board  *
;* Target MCU  : Atmel ATmega16                        *
;* Frequency   : X-TAL : 16 MHz                        *
;* Compiler    : AVR Studio 4.12 (AVR Assembler 2)     *
;* Create By   : ADISAK CHOOCHAN (WWW.ETT.CO.TH)       *
;* Last Update : 1/September/2005                      *
;* Description : Example LED Blink on Portb.0          *
;*******************************************************

;Connect PB0 to LED1

.include "m64def.inc"

;************************
; Define Register
;************************
.def      TEMP      = R16

;************************
; Define I/O Port,Pin
;************************
.equ      LED  =    0

;********************
; Main Program
;********************
.CSEG
          .ORG   0
          RJMP  RESET               ;Reset Handle

RESET:    LDI  TEMP,LOW(RAMEND)     ;Initial Stack Pointer
          OUT  SPL,TEMP
          LDI  TEMP,HIGH(RAMEND)
          OUT  SPH,TEMP

          SBI  DDRB,LED             ;Config Portb.0 as output

MAIN:     SBI  PORTB,LED            ;LED Off
          CBI  PORTB,LED            ;LED On
          RJMP MAIN                 ;Loop
```

6. Translate written program, click Menu **Build → Build** and after translated program correctly without any mistake, it will display message of **0 error 0 warnings** as in the picture.

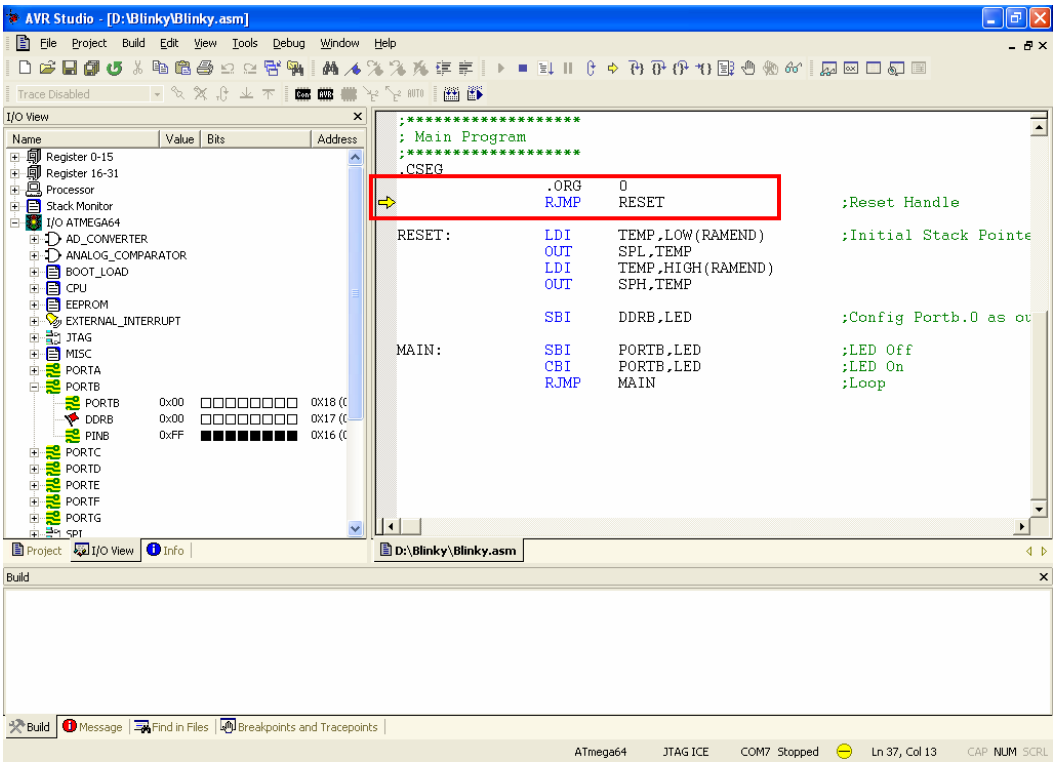



7. Click Menu instruction of **Debug → Start Debugging** as in the picture.

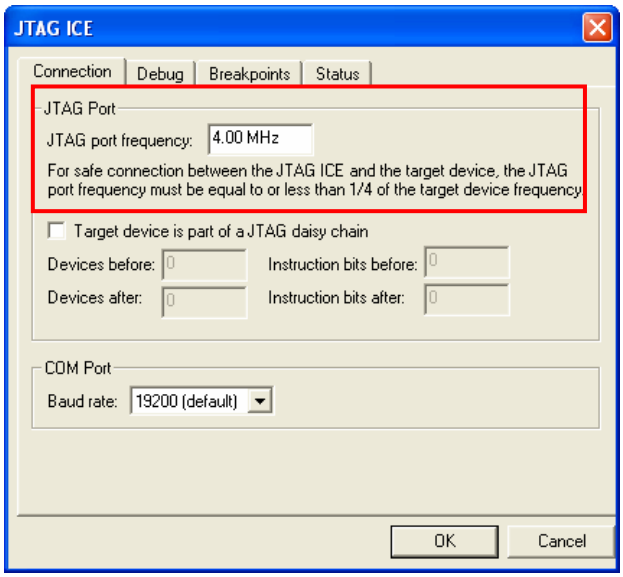



8. Program will download data into MCU and enter into debugging, there's a sign at the beginning of program as in the picture. On the right hand, it will display the window of **I/O**

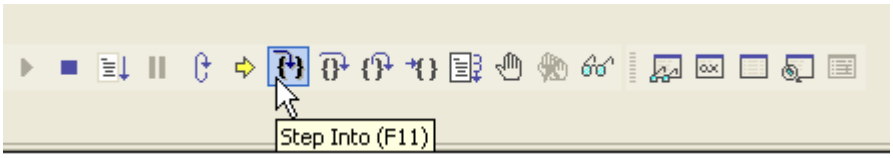
**View** that shows the default value of MCU Register.



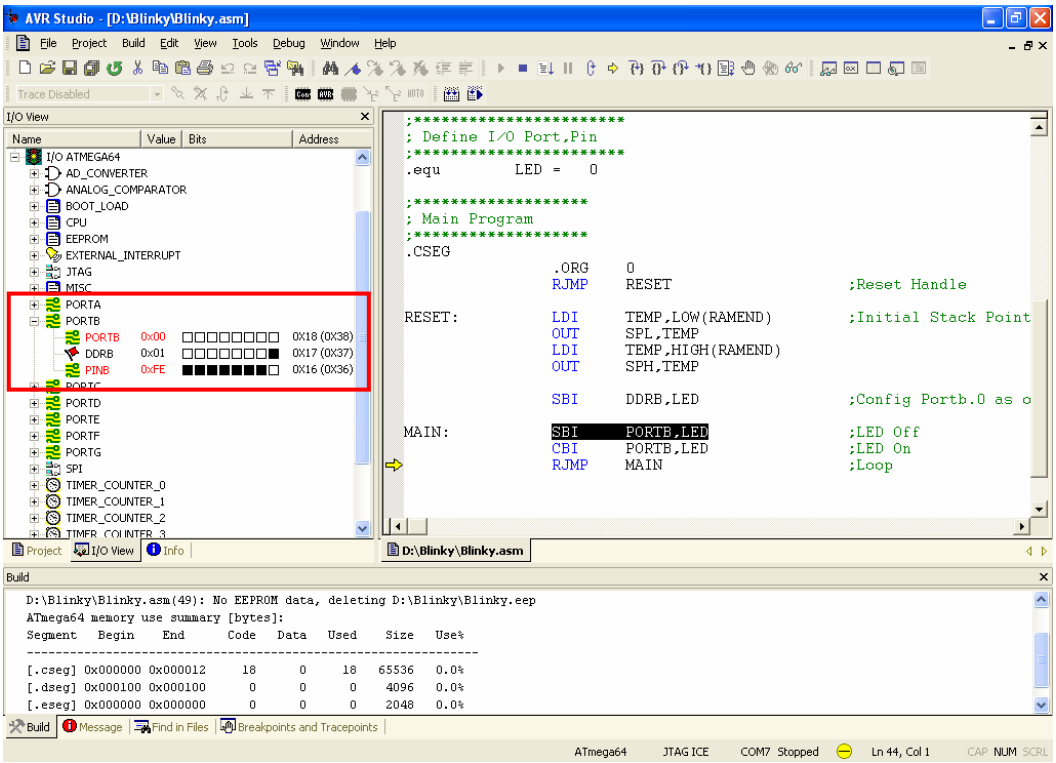

9. We can adjust speed of debugging by selecting form Menu instruction of **Debug → JTAG ICE Options** and can adjust **frequency of Port JTAG** that is not greater than 1/4 of using frequency.



10. Select Mode of Debugging, can select them as both **Auto Step** or **Step by Step.** Select from Material tab as in the picture, in this case, we select as **Step**.



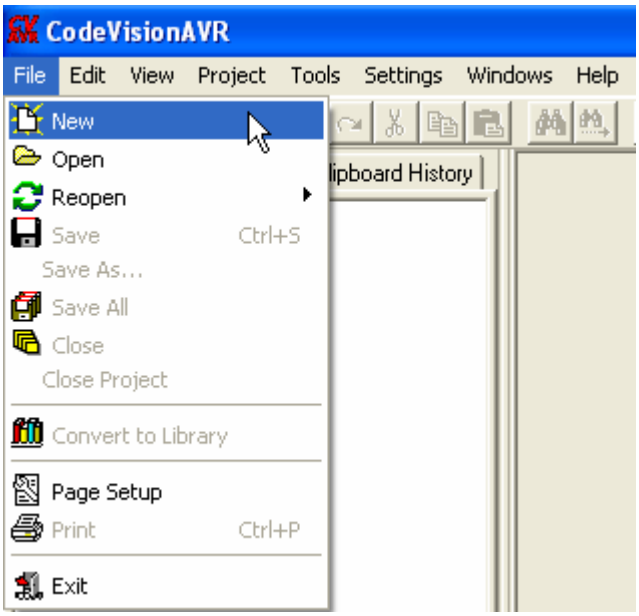

11. While debugging program through instruction of **SBI PORTB,LED** that is an instruction for assignment PORTB.0 as Logic 1 and LED at PORTB.0 is in status of switch off because board circuit assigns LED run with Logic 0. Press **Setup Info** again while debugging program through Instruction of **CBI PORTB,LED.** LED is in status of switch on and default values in the window of I/O View will change as programming.



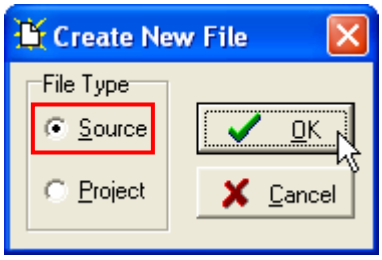
# Example of Debugging with C Language

Program AVR Studio can use with Assembly Language. Moreover, it can debug written C Language Software. In this case, we would like to describe the method to debug with C language by using Program CodeVisonAVR C complier with Program AVRStudio for debugging.

1. Open **Program CodeVisionAVR C Complier** and then click Menu instruction of **File → New** as in the picture.



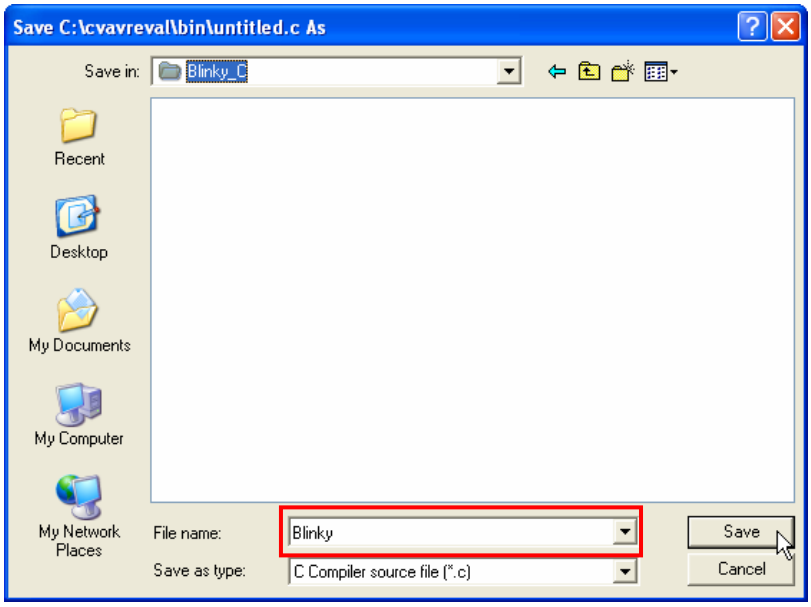

2. Select **File Type** as **Source** to create new C Language File and then click **OK** as in the picture.



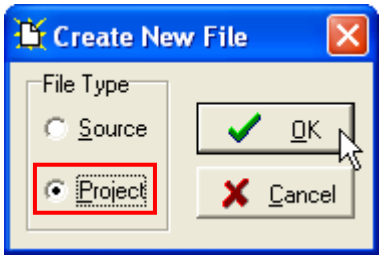
3. It will display window of **Editor** for writing program as in the sample.

```
//*********************************************/;
//*Hardware    : ET-AVR STAMP (ATmega64)       */;
//*CPU         : ATMEL-ATmega64                */;
//*X-TAL       : 16.00 MHz                     */;
//*Filename    : Main.C                        */;
//*Complier    : CodeVisionAVR V1.24.7d        */;
//*Last Update : 9-12-2005(ETT CO.,LTD)        */;
//*            : WWW.ETT.CO.TH                 */;
//*Description : Example LED Blink on Portb.0  */;
//*********************************************/;
//*CodeVisionAVR Complier Option Setting       */;
//*Chip type             : ATmega64            */;
//*Program type          : Application         */;
//*Clock frequency       : 16.000000MHz        */;
//*Memory model          : Small               */;
//*External SRAM size    : 0                   */;
//*Data Stack size       : 1024                */;
//*********************************************/;

#include <mega64.h>            // ATmega64 MCU
#include <delay.h>             // Delay functions

void main(void)
{
  PORTB=0x00;                  // PB7..0 = 0
  DDRB=0x01;                   // PB0 = Output

 //Loop Blink LED on PB0
  while (1)
  {
    PORTB |= 0x01;             // PB0 = 1 (OFF LED)
    PORTB &= 0XFE;             // PB0 = 0 (ON LED)
  }
}
```

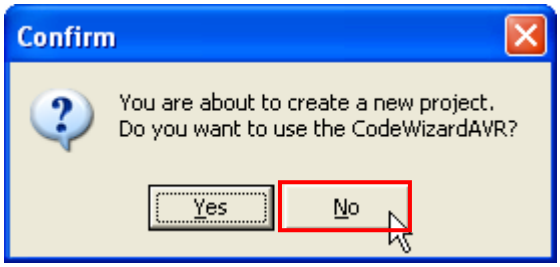
4. Save C Language program. Click Menu **File → Save** and assign File name and then click **Save** as in the picture.



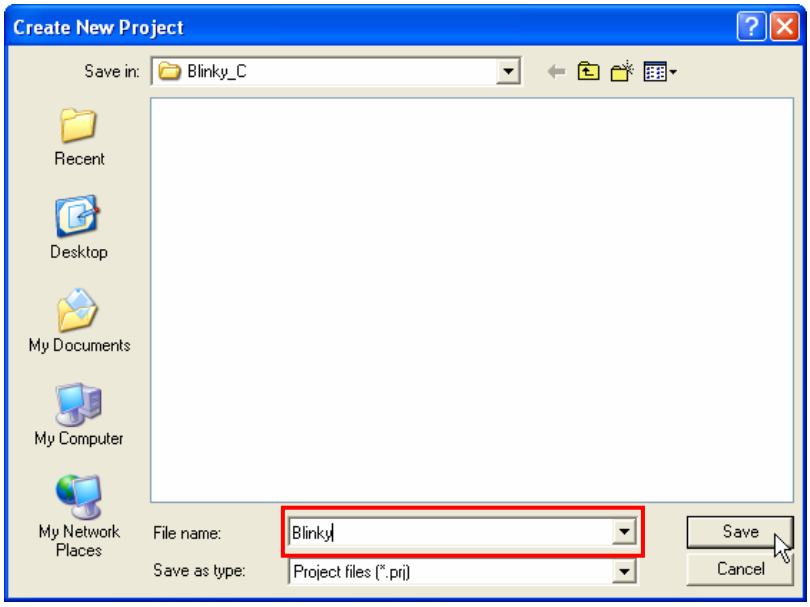

5. Select Menu **File → New** and then select **File Type** as **Project** to create new project and then click **OK** as in the picture.



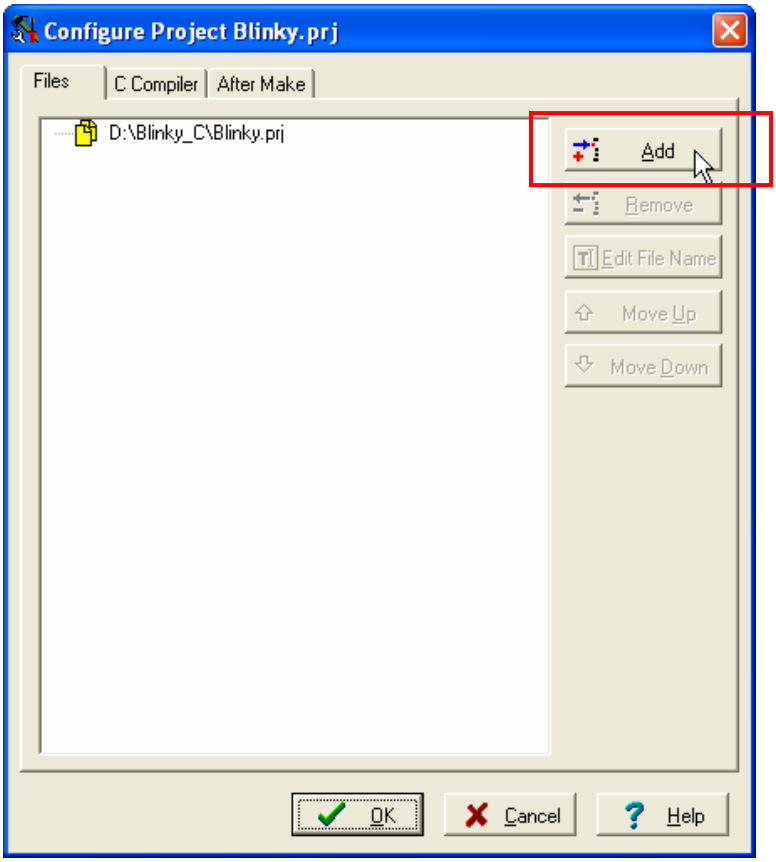

6. Click **No** because we do not need Help to create project (CodeWizard).

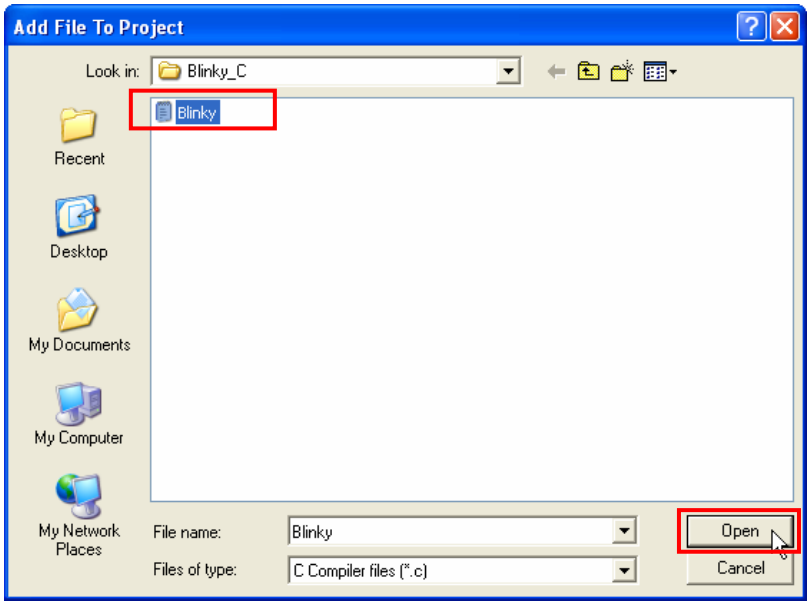


7. Assign Project Name as required and then click **Save** as in the picture.

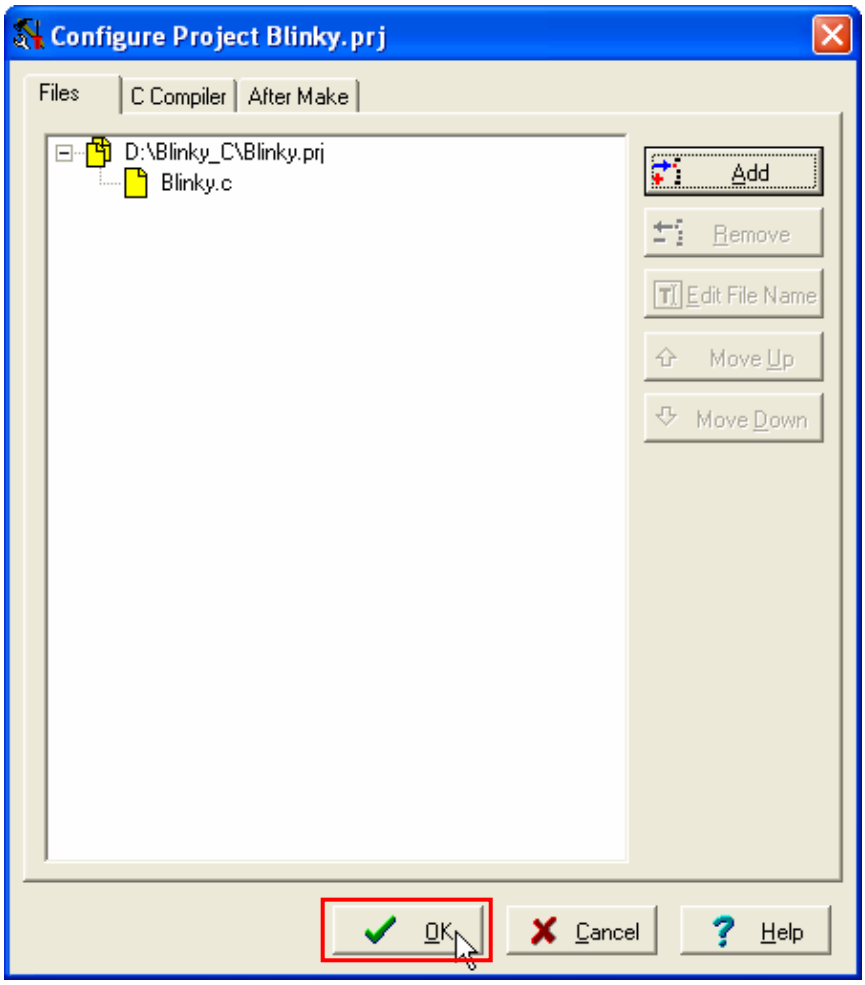



8. Add written C Language File into project, click **Add** as in the picture.

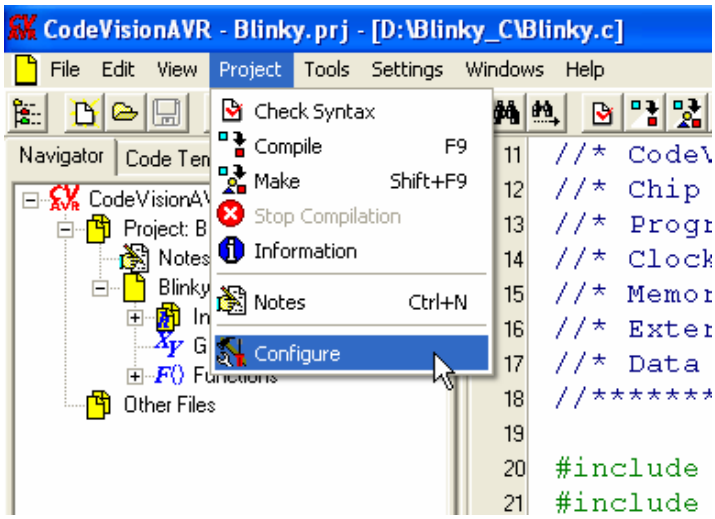


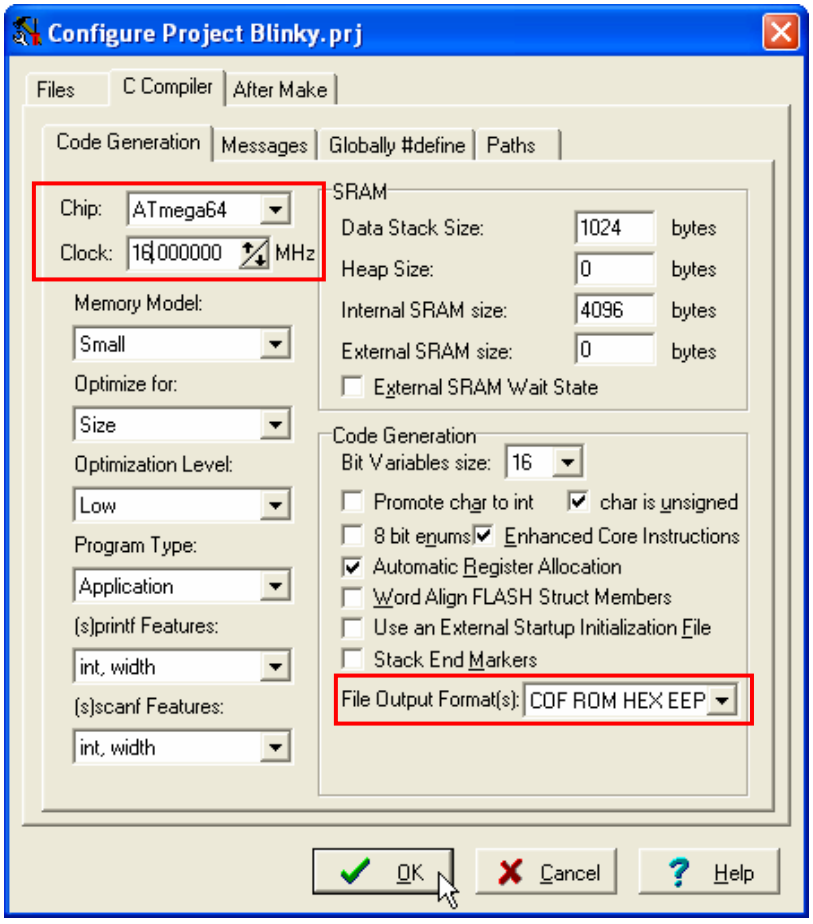



9. Lastly, click **OK** as in the picture.

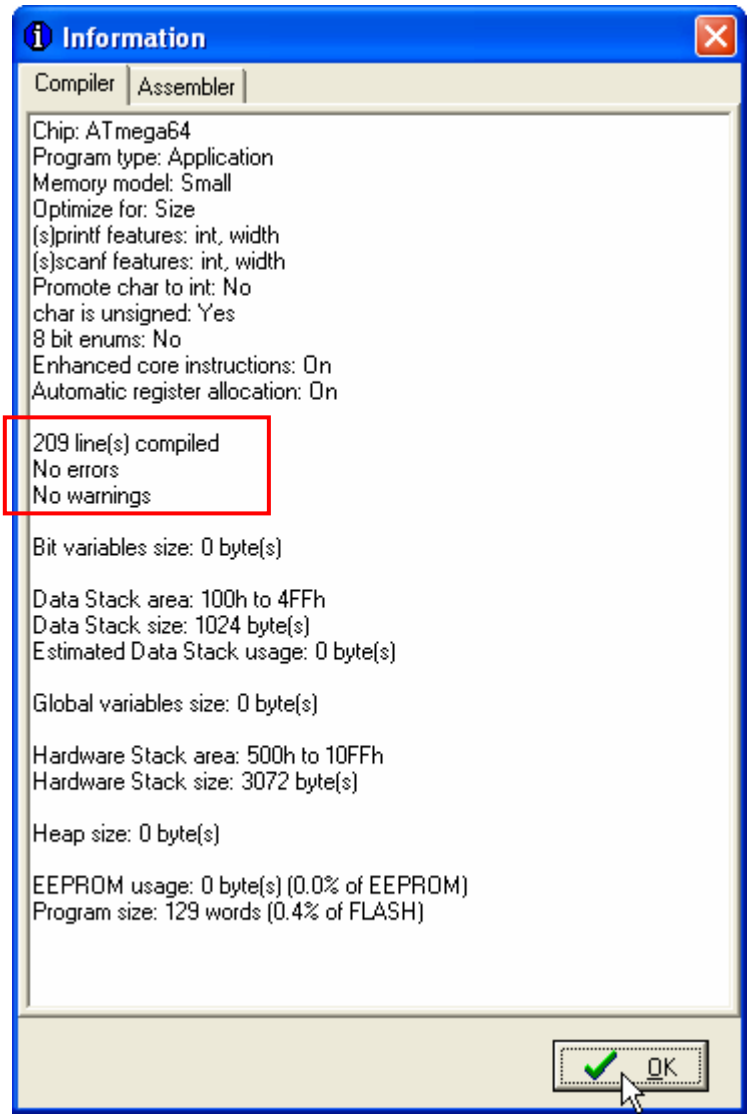


**10.** Assign default values of project, click Menu instruction of **Project → Configure** and then assign **MCU No.** as **ATmega64**, **Crystal** as **16.000000MHz** and **File Output Format(**s) as **COF ROM HEX EEP.**

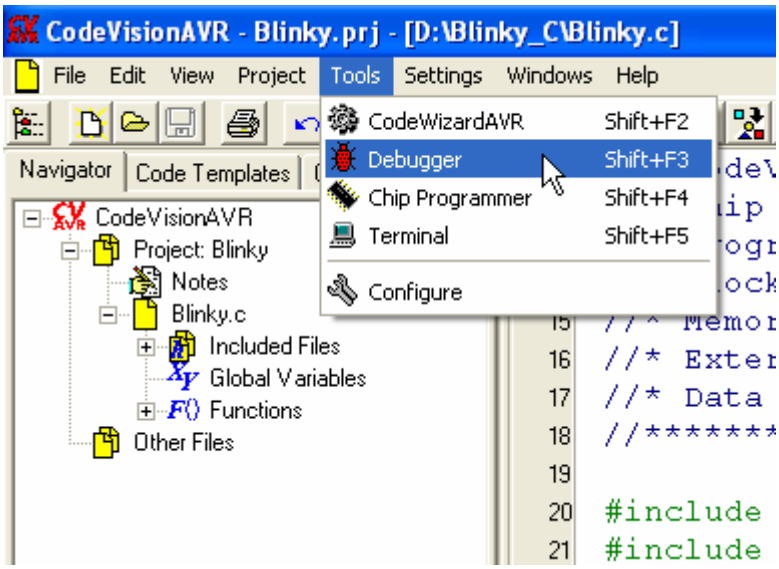


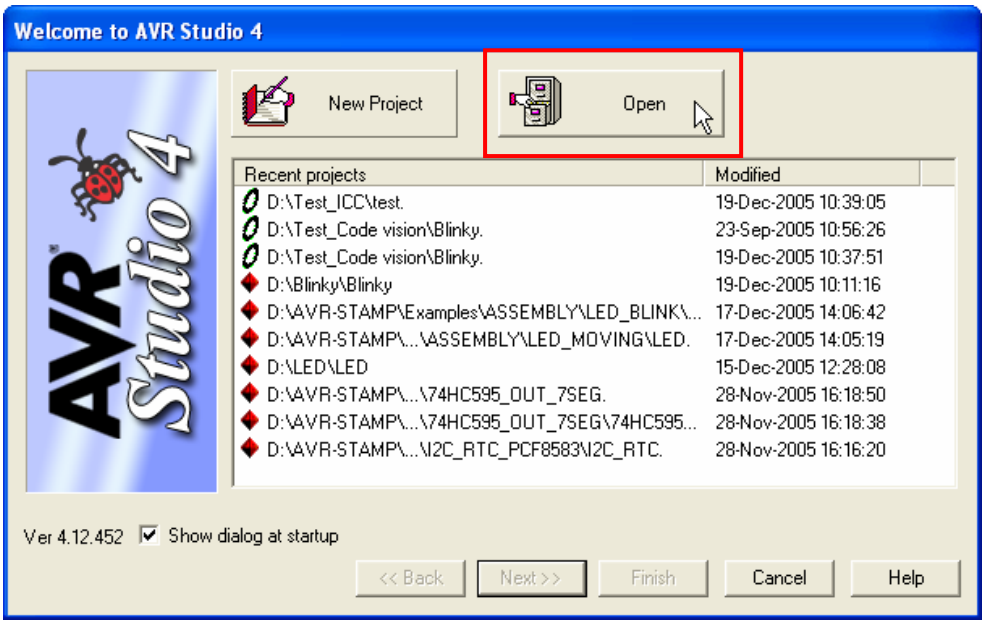



**11.** Translate written program by clicking Menu Instruction **Project → Make**. After programming completely, if there's no any mistake, it will display message as **No errors, no warnings** as in the picture.

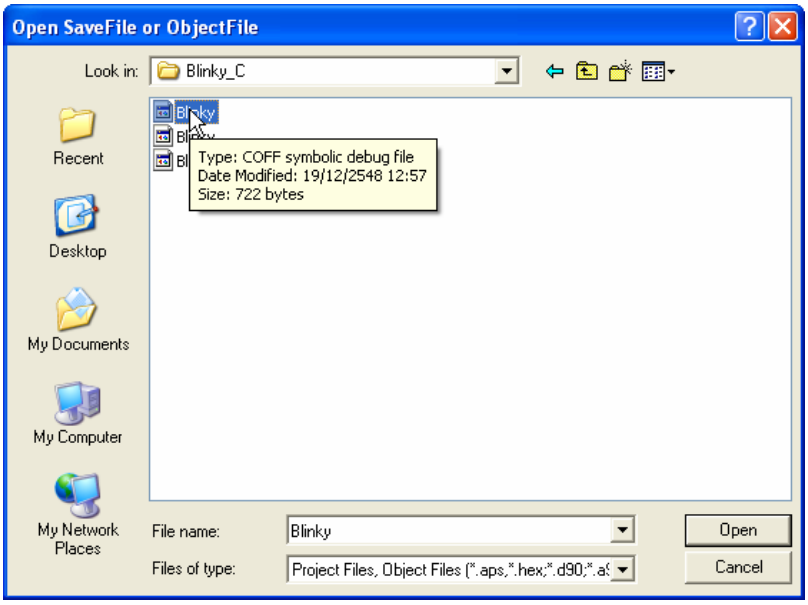


12. Click Menu Instruction of **Tools → Debugger** for entering into debugging program and after that program CodeVisionAVR will open program AVR Studio as in the picture. Click **Open**.

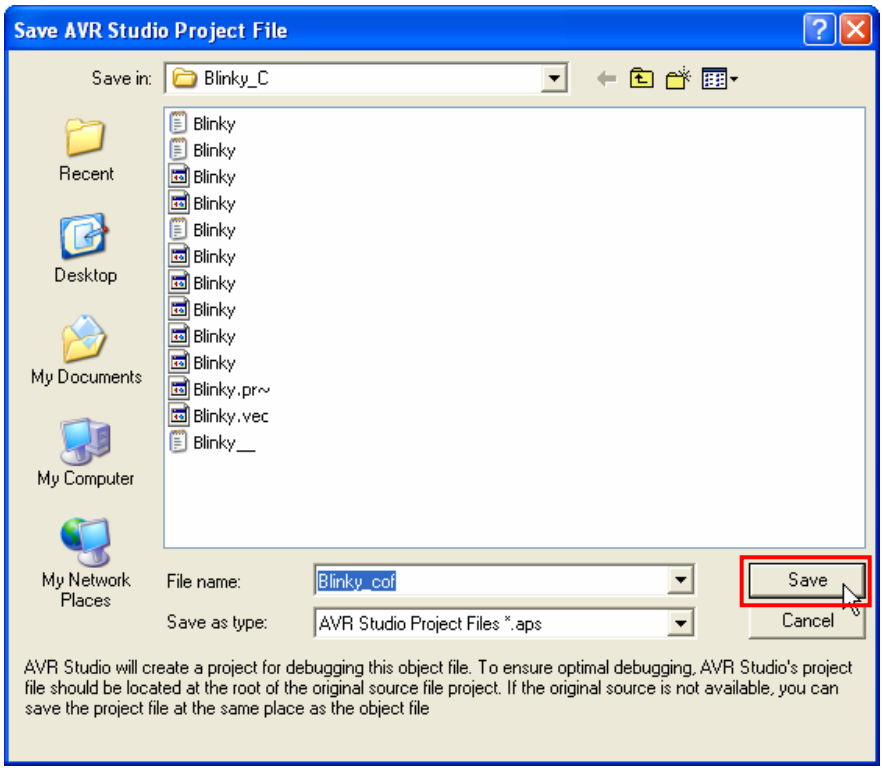


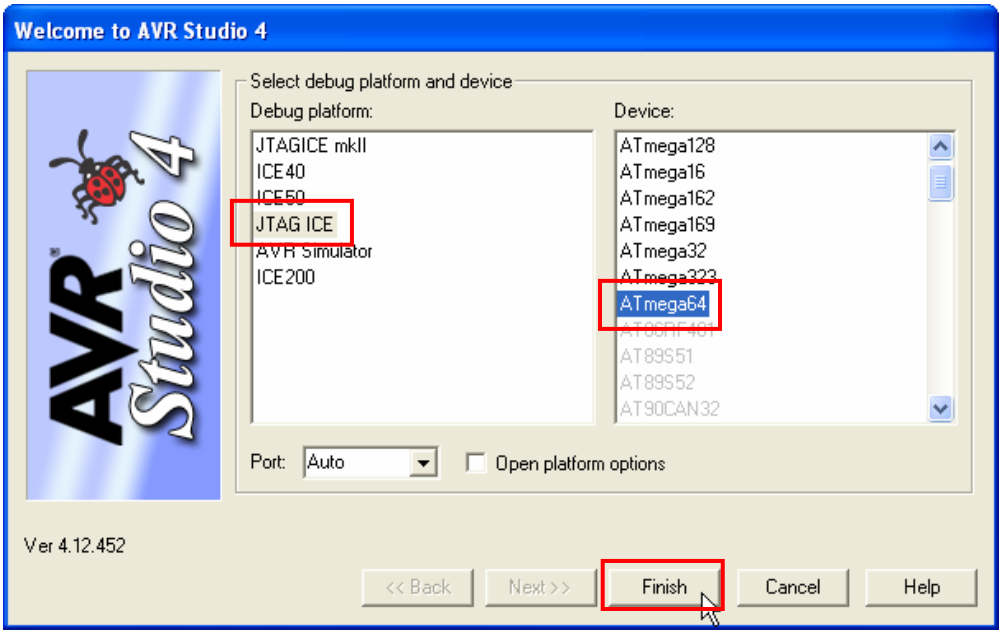
13. Select File **COFF symbolic debug file** that is translated from created project as in the picture.



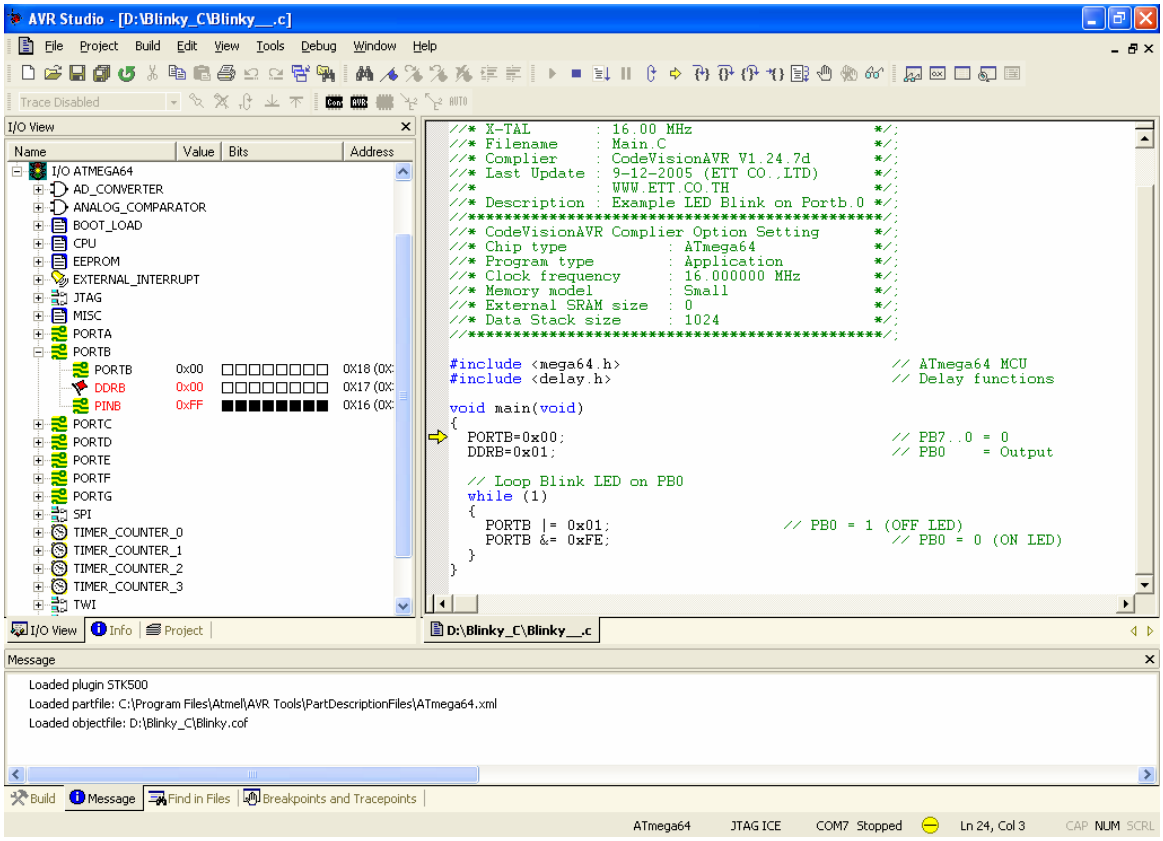

14. Then program will save project , click **Save**.



15. Select **Debug platform** as **JTAG ICE** and **Device** as **ATmega64** and then click **Finish** as in the picture.



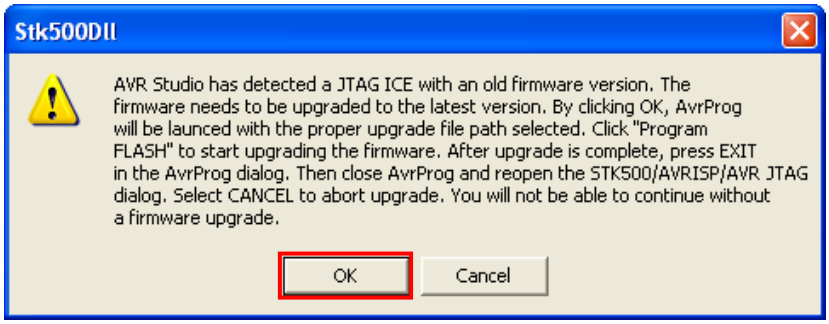

16. When it is completely, pogram AVR Studio will download written C Language program and after that user can debug program as same as debug program of Assembly Language.



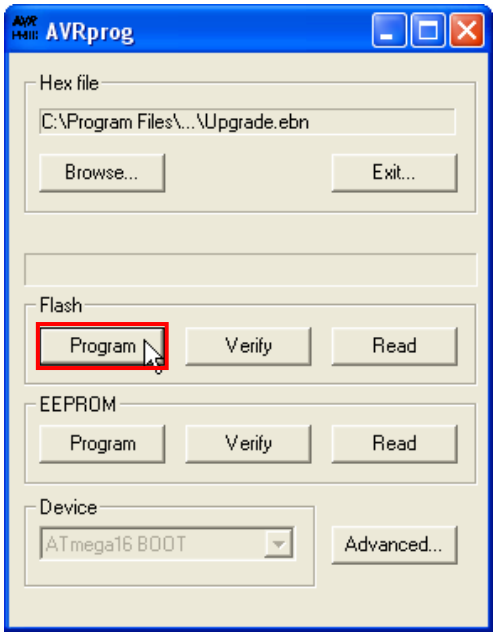

## Proceeding to Upgrade Firmware of ET-JTAG AVR

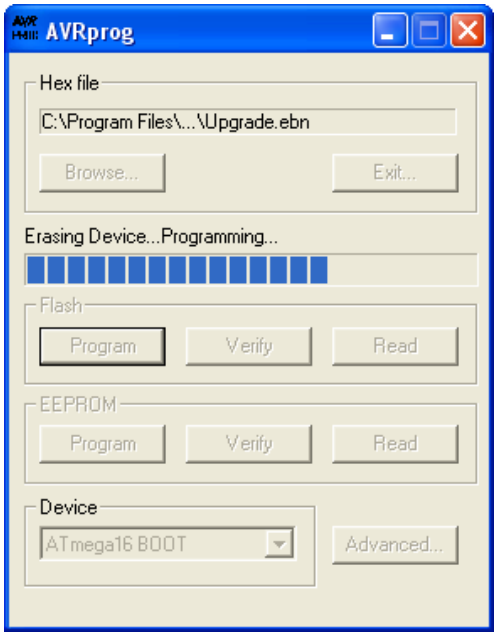
To upgrade firmware, it makes ET-JTAG AVR can use with new MCU No.This Firmware is attached with program AVR Studio and when user connects ET-JTAG AVR with Program AVR Studio, if there's new Fiemware, it will display the message to upgrade new Firmware. The method to upgrade Firmware is;

1. When connection with ET-JTAG AVR and there' s new firmware, it will display message as in the picture. Click **OK** to upgrade new Firmware.

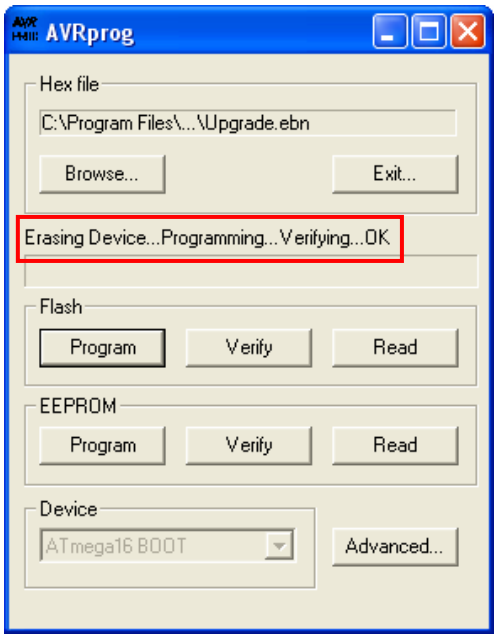



2. Click **Browse** to select file **Upgrade.ebn.**from program AVR Prog. Normally, it is in Directory **C:\Program Files\Atmel\AVR Tools\JTAGICE** (Normally, program will select automatically). And then click **Program** to start upgrade as in the picture.





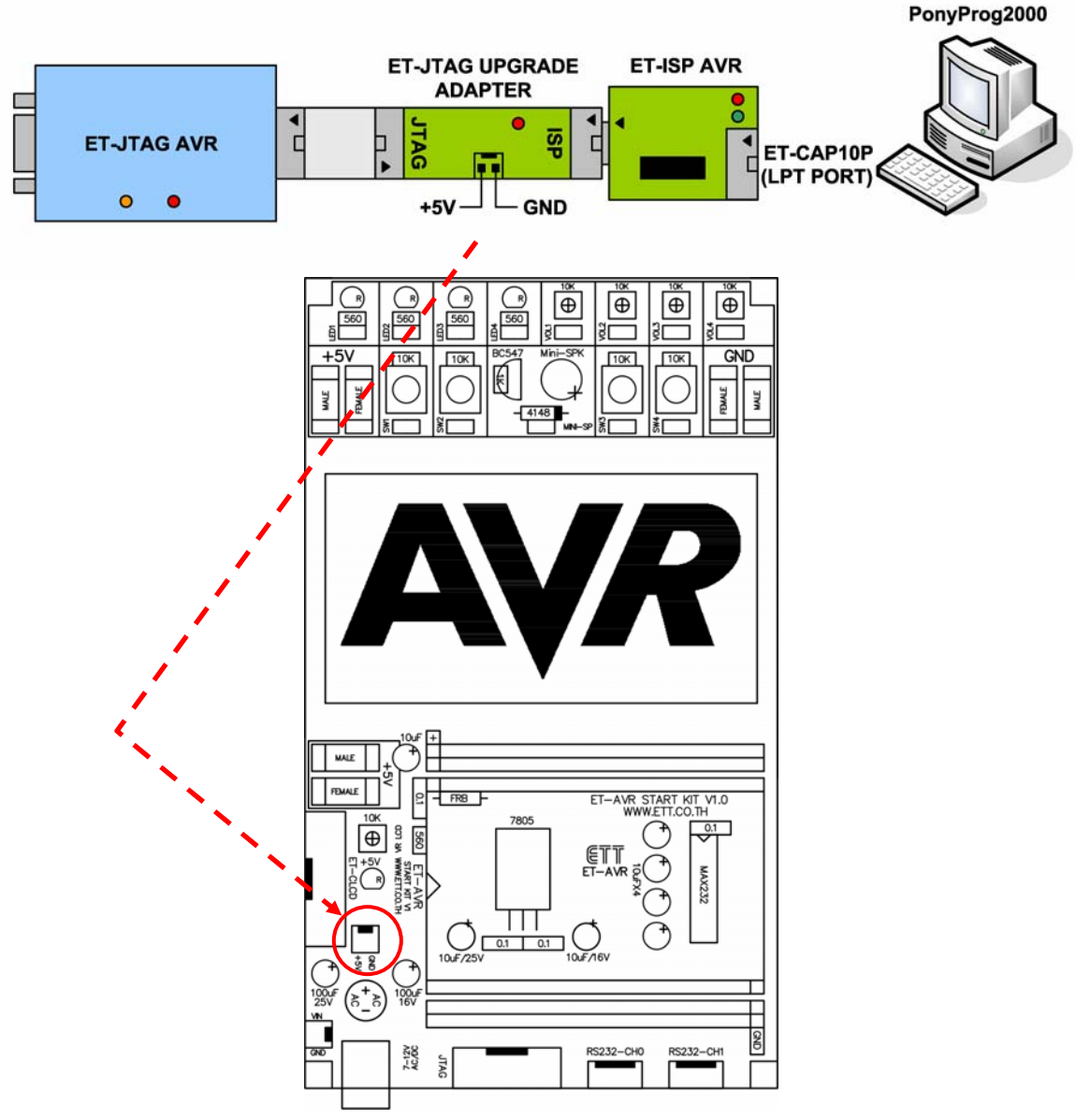

3. Close program AVR Prog when it is completely. Then take off all Power Supply of ET-JTAG AVR and now new Firmware is upgraded completely.



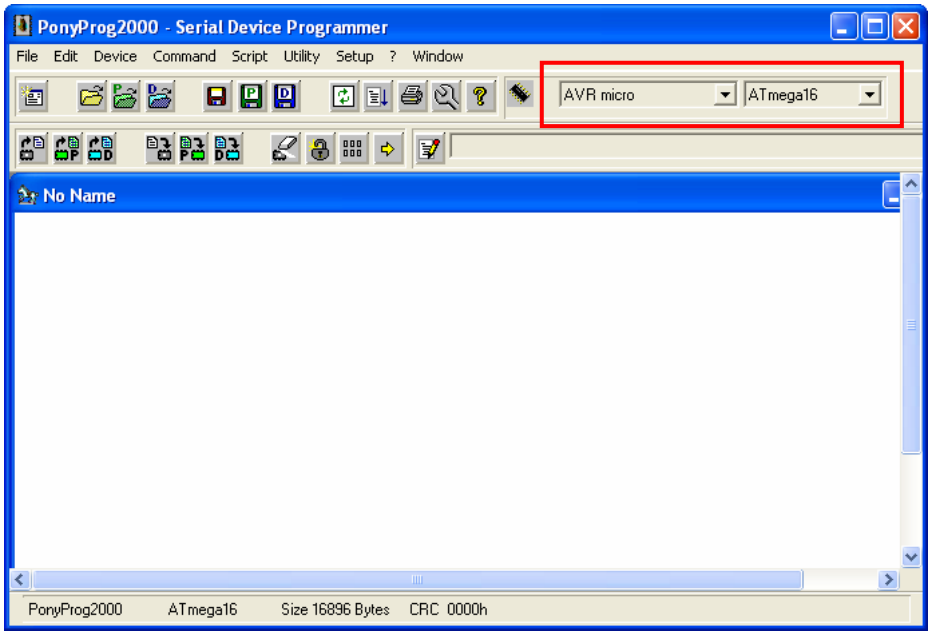

## NOTICE

In case of can not auto upgrade new Firmware, it may be problem with USB TO SERIAL. So, user can upgrade with **ET-JTAG UPGARDE ADATER** and **ET-ISP AVR** as following.

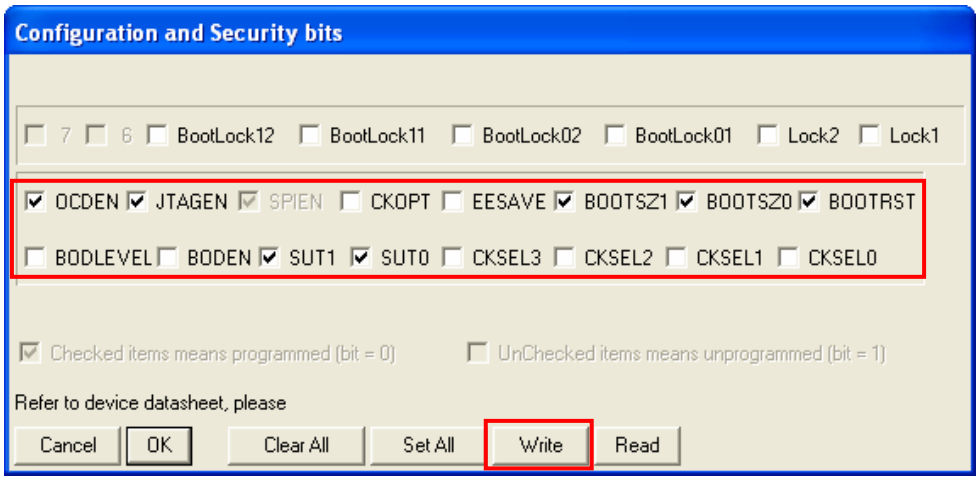
1. Interface ET-JTAG AVR with ET-JTAG UPGRADE ADAPTER and ET-ISP AVR by connecting ET-ISP AVR with Parallel Port of computer and using ET-CAP10P as in the picture. Because ET-JTAG AVR uses power supply from external, so it must receive 5V power supply through ET-JTAG UPGRADE ADAPTER and user can interface 5V power supply from Board ET-AVR START KIT V1 because there's a connector to interface.



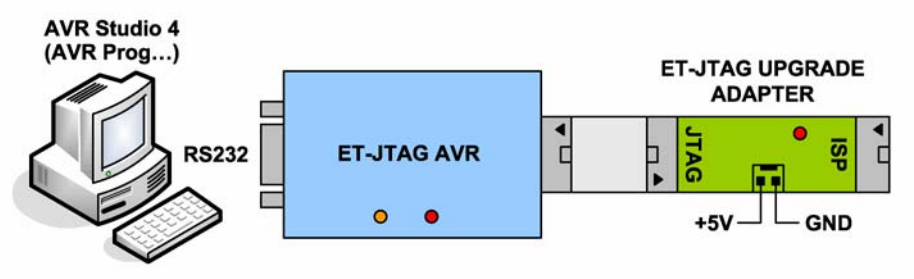
2. Open program PonyProg2000 and then assign CPU No. from **Device → AVR Micro →** as **ATmega16.**



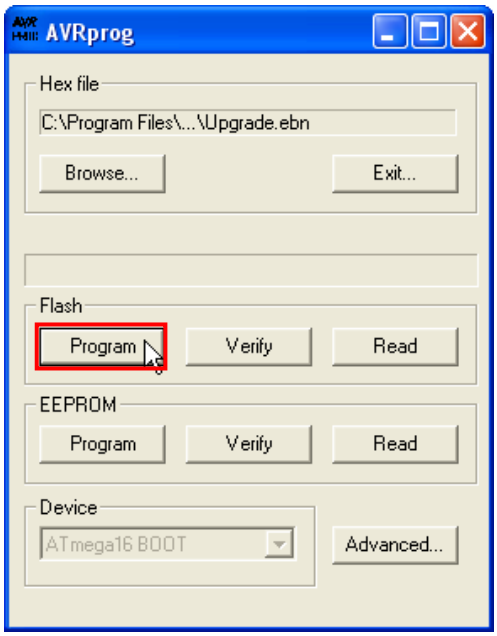

3. Select Menu **Command → Security and Configuration Bits** and then select Fuse bit as in the picture. Click **Write**.

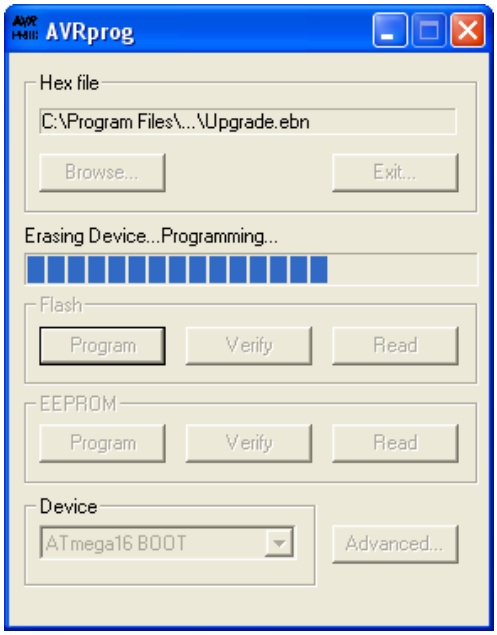



4. Take off 5V Power Supply from Board ET-JTAG UPGRADE ADAPTER and take off ET-ISP AVR from ET-JTAG UPGRADE ADAPTER. Interface ET-JTAG AVR with computer through RS232 Port and interface 5V Power Supply again as in the picture.

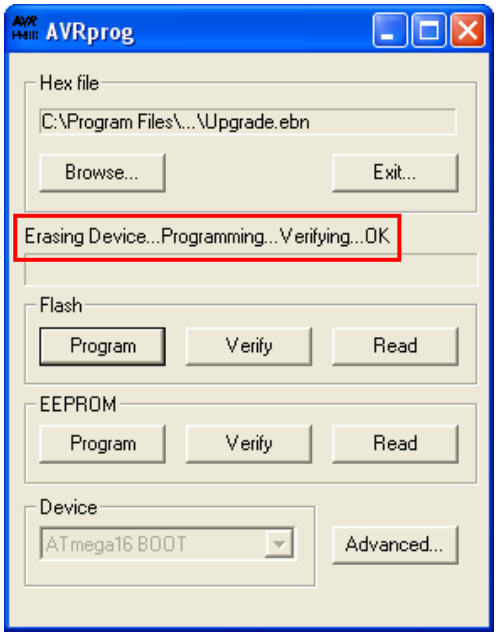



5. Open program AVR Studio 4 and select Menu **Tools → AVR Prog…** and then click **Browse** to select file Upgrade.ebn. Normally, it is in directory **C:\Program Files\Atmel\AVR Tools\JTAGICE** (Normally, program selects file automatically). Then click **Program** to start upgrade as in the picture.

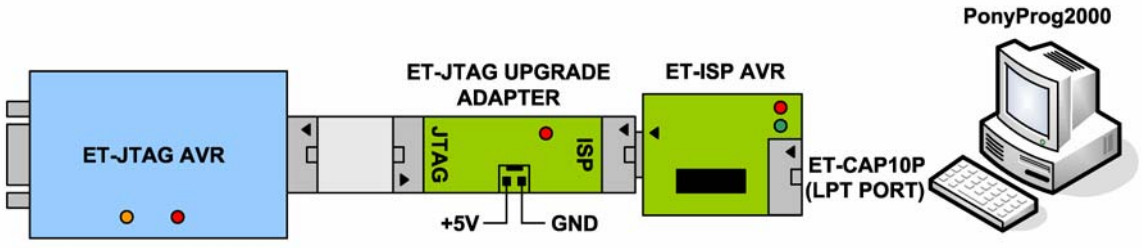





6. Close Program AVR Prog when it is completely. Then take off 5V Power Supply from Board ET-JTAG UPGRADE ADAPTER.




7. Interface circuit as same as in No.1 and take off ET-JTAG AVR from computer. After that interface 5V Power Supply into Board ET-JTAG UPGRADE ADAPTER.




8. Open program PonyProg2000 and then select Fuse Bit as in the picture. Click **Write** and now new Firmware is upgraded completely.